

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятности

ОТЧЕТ
по лабораторной работе
ТЕМА «Управление версиями»
по дисциплине «Операционные системы»

Выполнил:

Студент группы НПИбд-02-21

Щербак Маргарита Романовна

«23» апреля 2022г.

Москва 2022

Цель работы: Изучить идеологию и применение средств контроля версий. — Освоить умения по работе с git.

Ход работы:

Из теории:

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями. Управление версиями Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

Настройка github 1.

1.Создали учётную запись на <https://github.com>.

2. Заполнили основные данные на <https://github.com>. (Рис.1)

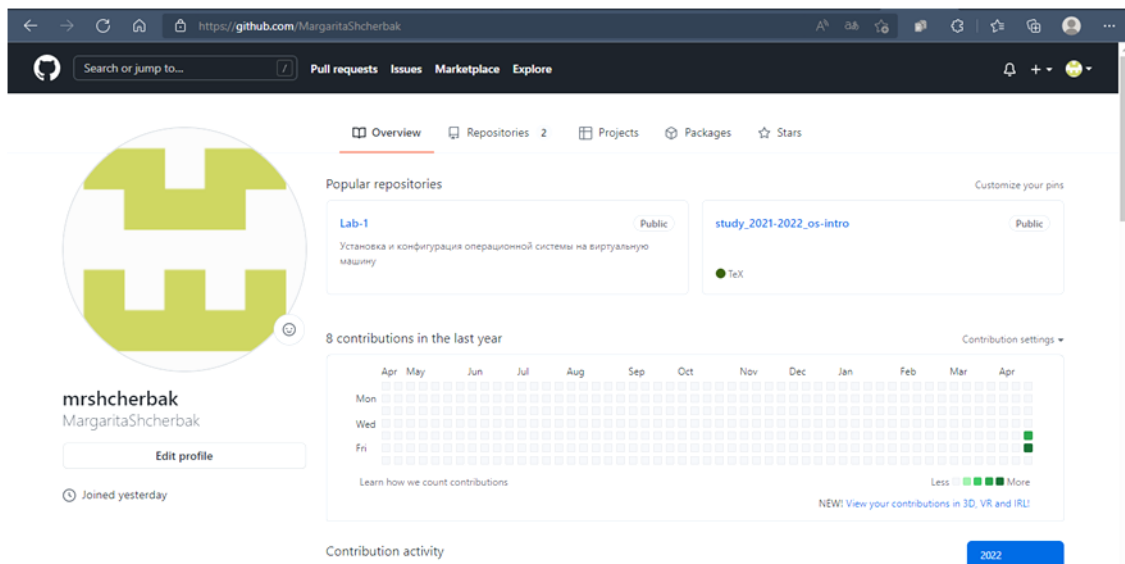


Рис.1

Установка программного обеспечения.

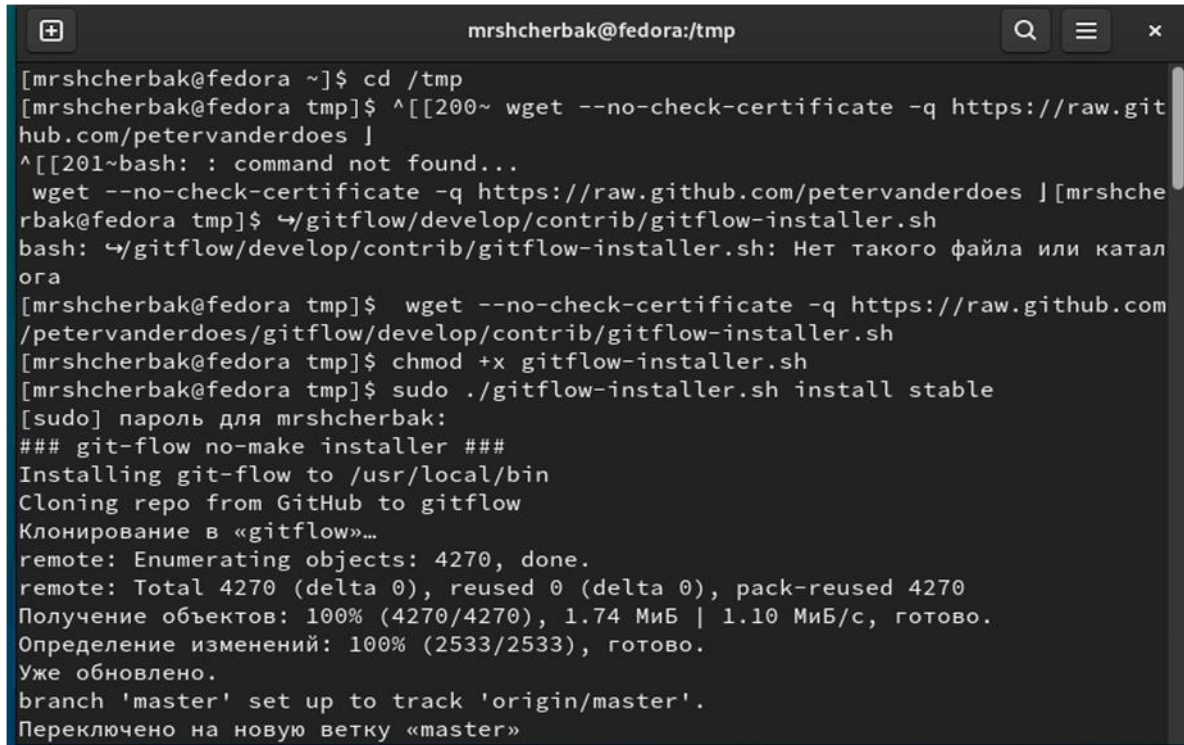
Установка git-flow в Fedora Linux – это программное обеспечение удалено из репозитория. – Необходимо устанавливать его вручную. (Рис.2)

Ввеликоманды:

```

cd /tmp
wget --no-check-certificate -q
https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
chmod +x gitflow-installer.sh
sudo ./gitflow-installer.sh install stable

```



```

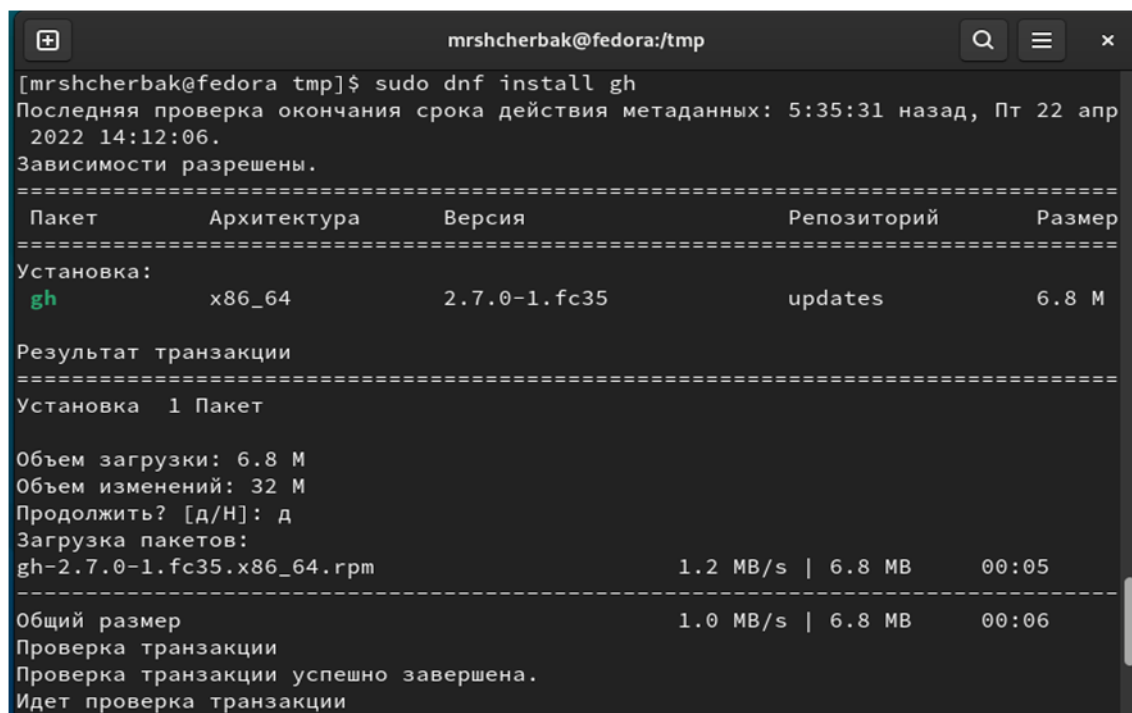
mrshcherbak@fedora:~/tmp
[mrshcherbak@fedora ~]$ cd /tmp
[mrshcherbak@fedora tmp]$ ^[[200~ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes ]
^[[201~bash: : command not found...
[mrshcherbak@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes ] [mrshcherbak@fedora tmp]$
[mrshcherbak@fedora tmp]$ ./gitflow/develop/contrib/gitflow-installer.sh
bash: ./gitflow/develop/contrib/gitflow-installer.sh: Нет такого файла или каталога
[mrshcherbak@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[mrshcherbak@fedora tmp]$ chmod +x gitflow-installer.sh
[mrshcherbak@fedora tmp]$ sudo ./gitflow-installer.sh install stable
[sudo] пароль для mrshcherbak:
### git-flow no-make installer ###
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Клонирование в «gitflow»...
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270
Получение объектов: 100% (4270/4270), 1.74 МиБ | 1.10 МиБ/с, готово.
Определение изменений: 100% (2533/2533), готово.
Уже обновлено.
branch 'master' set up to track 'origin/master'.
Переключено на новую ветку «master»

```

.Рис.2

Установка gh в FedoraLinux. (Рис.3)

Ввели команду `sudo dnf install gh`

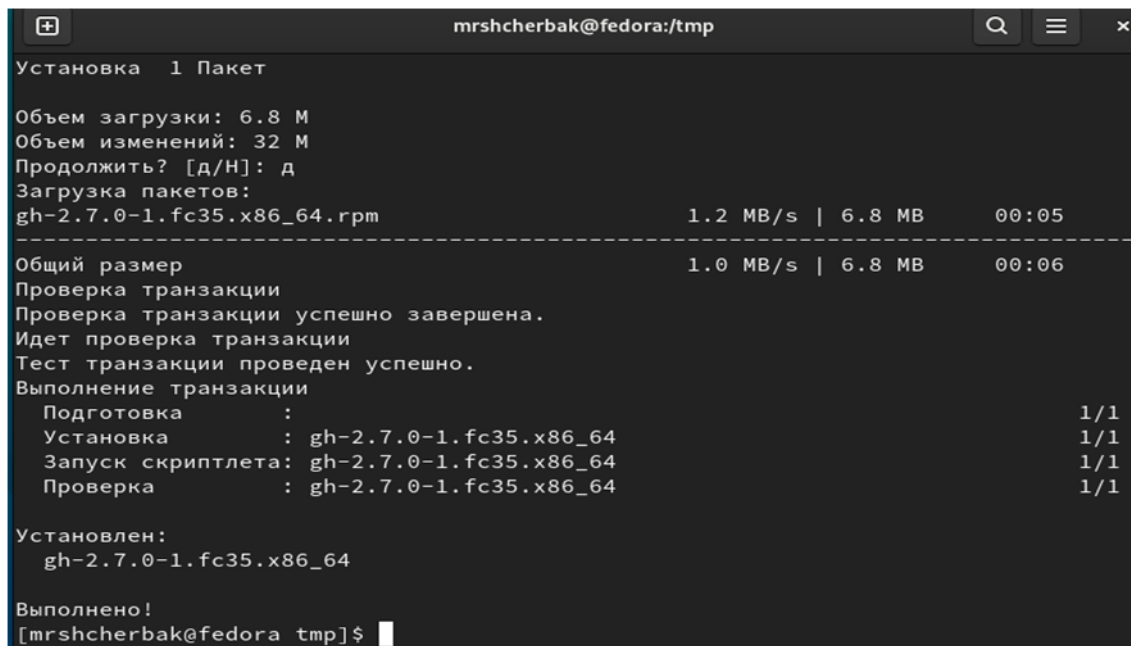


```

mrshcherbak@fedora:~/tmp
[mrshcherbak@fedora tmp]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 5:35:31 назад, Пт 22 апр 2022 14:12:06.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh          x86_64       2.7.0-1.fc35 updates      6.8 М
=====
Результат транзакции
=====
Установка  1 Пакет

Объем загрузки: 6.8 М
Объем изменений: 32 М
Продолжить? [д/Н]: д
Загрузка пакетов:
gh-2.7.0-1.fc35.x86_64.rpm      1.2 MB/s | 6.8 MB  00:05
-----
Общий размер                    1.0 MB/s | 6.8 MB  00:06
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции

```



```
Установка 1 Пакет
Объем загрузки: 6.8 М
Объем изменений: 32 М
Продолжить? [д/Н]: д
Загрузка пакетов:
gh-2.7.0-1.fc35.x86_64.rpm 1.2 MB/s | 6.8 MB 00:05
-----
Общий размер 1.0 MB/s | 6.8 MB 00:06
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка : 1/1
Установка : gh-2.7.0-1.fc35.x86_64 1/1
Запуск скрипглета: gh-2.7.0-1.fc35.x86_64 1/1
Проверка : gh-2.7.0-1.fc35.x86_64 1/1

Установлен:
gh-2.7.0-1.fc35.x86_64

Выполнено!
[mrshcherbak@fedora tmp]$
```

Рис.3

Базовая настройка git Рис.4

Задали имя и email владельца репозитория:

```
git config --global user.name "Name Surname"
```

```
git config --global user.email "work@mail"
```

Настроим utf-8 в выводе сообщений git:

```
git config --global core.quotePath false
```

 – Настроили верификацию и подписание коммитов git. Задали имя начальной ветки (будем называть её master)

```
gitconfig --globalinit.defaultBranchmaster
```

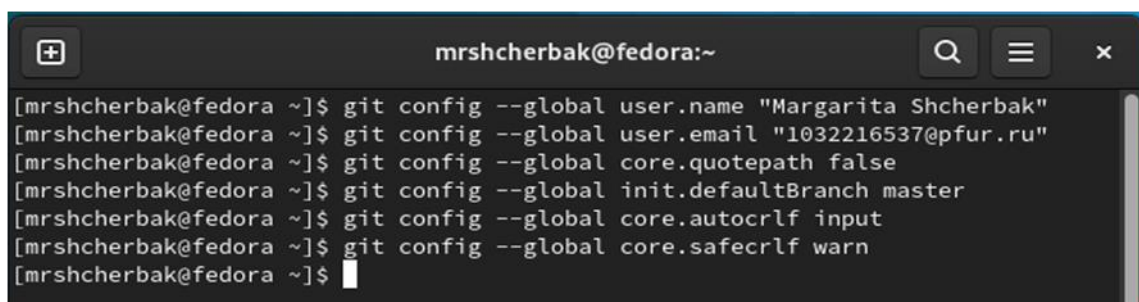
Параметр autocrlf:

```
git config --global core.autocrlf input
```

Параметр safecrlf:

```
git config --global core.safecrlf warn
```

Почту изменила на batyalove13@gmail.com как на Github.



```
[mrshcherbak@fedora ~]$ git config --global user.name "Margarita Shcherbak"
[mrshcherbak@fedora ~]$ git config --global user.email "1032216537@pfur.ru"
[mrshcherbak@fedora ~]$ git config --global core.quotePath false
[mrshcherbak@fedora ~]$ git config --global init.defaultBranch master
[mrshcherbak@fedora ~]$ git config --global core.autocrlf input
[mrshcherbak@fedora ~]$ git config --global core.safecrlf warn
[mrshcherbak@fedora ~]$
```

Рис.4

Создали ключи ssh

– по алгоритму rsa с ключём размером 4096 бит:

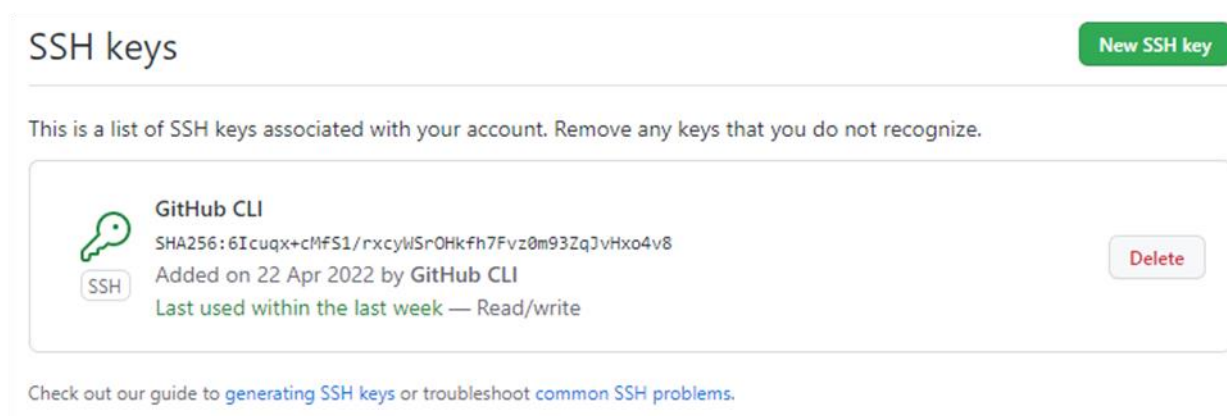
```
ssh-keygen -t rsa -b 4096
```

– поалгоритму ed25519:

```
ssh-keygen -t ed25519
```

```
mrshcherbak@fedora:~  
+-----[SHA256]-----+  
[mrshcherbak@fedora ~]$ ssh-keygen -t ed25519  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/mrshcherbak/.ssh/id_ed25519): id_ed25519  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in id_ed25519  
Your public key has been saved in id_ed25519.pub  
The key fingerprint is:  
SHA256:HlrdIMaR7gqcK7OR320jm4gsrD1D/lfcY1N859s9Las mrshcherbak@fedora  
The key's randomart image is:  
+---[ED25519 256]---+  
|      ..      |  
|      ...     |  
|     .+  ..   |  
|    ..o oo . . |  
|   . . oS.....o |  
|  ..+  ++.=   . |  
|.oo  o.o.. o   =|  
|.+++.o+oo     .o+|  
|o.=B+.++...  E..o.|  
+-----[SHA256]-----+  
[mrshcherbak@fedora ~]$
```

Рис.5



Создали ключи pgr Рис.6

– Генерируем ключ

gpg --full-generate-key

Из предложенных опций выбираем:

– тип RSA and RSA;

– размер 4096;

– выберите срок действия; значение по умолчанию — 0 (срок действия не истекает никогда).

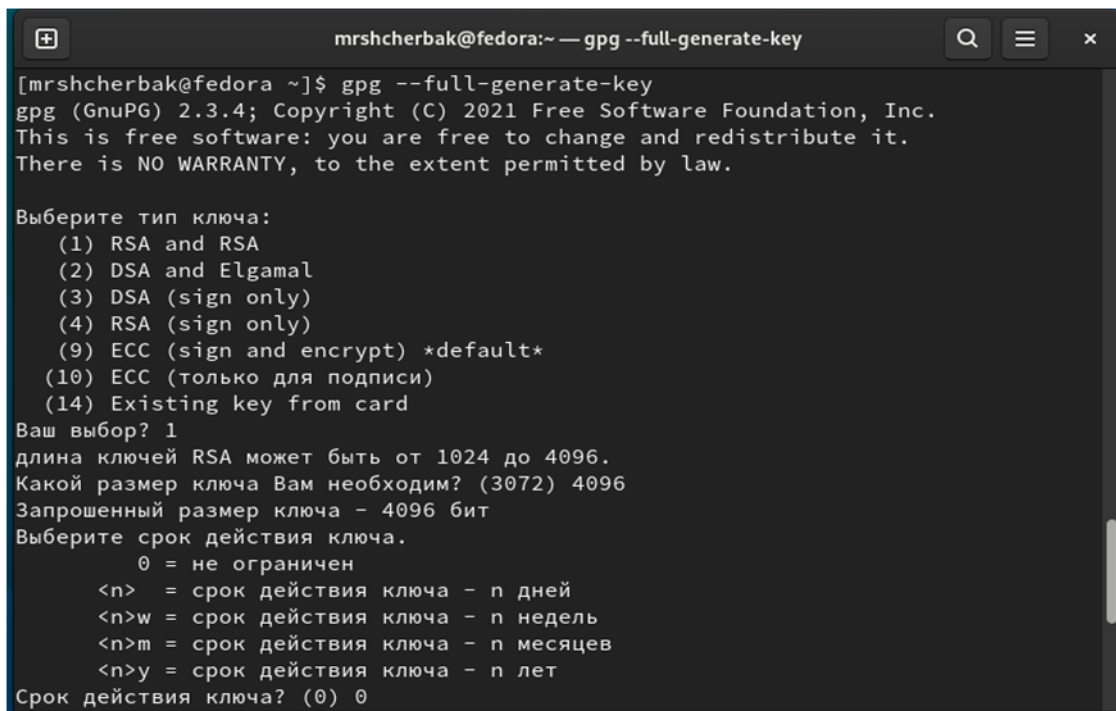
– GPG запросит личную информацию, которая сохранится в ключе:

– Имя (не менее 5 символов).

– Адрес электронной почты.

– При вводе email убедитесь, что он соответствует адресу, используемому на GitHub.

– Комментарий. Можно ввести что угодно или нажать клавишу ввода, чтобы оставить это поле пустым.



```
mrshcherbak@fedora:~ — gpg --full-generate-key
[mrshcherbak@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.4; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (только для подписи)
 (14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
  0 = не ограничен
  <n> = срок действия ключа - n дней
  <n>w = срок действия ключа - n недель
  <n>m = срок действия ключа - n месяцев
  <n>y = срок действия ключа - n лет
Срок действия ключа? (0) 0
```

Рис.6

Добавление PGP ключа в GitHub

– Вывели список ключей и копировали отпечаток приватного ключа:

`gpg --list-secret-keys --keyid-format LONG`

– Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.

Формат строки:

`sec Алгоритм/Отпечаток_ключа Дата_создания [Флаги] [Годен_до] ID_ключа`

– Скопировали наш сгенерированный PGP ключ в буфер обмена:

`gpg --armor --export | xclip -sel clip`



– Перешли в настройки GitHub (<https://github.com/settings/keys>), нажали на кнопку New GPG key и вставили полученный ключ в поле ввода.

```
mrshcherbak@fedora:~  
"mrshcherbak (OC Lab02) <batyalove13@gmail.com>"  
  
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? O  
Необходимо получить много случайных чисел. Желательно, чтобы Вы  
в процессе генерации выполняли какие-то другие действия (печать  
на клавиатуре, движения мыши, обращения к дискам); это даст генератору  
случайных чисел больше возможностей получить достаточное количество энтропии.  
Необходимо получить много случайных чисел. Желательно, чтобы Вы  
в процессе генерации выполняли какие-то другие действия (печать  
на клавиатуре, движения мыши, обращения к дискам); это даст генератору  
случайных чисел больше возможностей получить достаточное количество энтропии.  
gpg: /home/mrshcherbak/.gnupg/trustdb.gpg: создана таблица доверия  
gpg: создан каталог '/home/mrshcherbak/.gnupg/openpgp-revocs.d'  
gpg: сертификат отзыва записан в '/home/mrshcherbak/.gnupg/openpgp-revocs.d/C6CE  
AB9E766FAFF54298109D4D44943B3728F788.rev'.  
открытый и секретный ключи созданы и подписаны.  
  
pub   rsa4096 2022-04-22 [SC]  
      C6CEAB9E766FAFF54298109D4D44943B3728F788  
uid    mrshcherbak (OC Lab02) <batyalove13@gmail.com>  
sub    rsa4096 2022-04-22 [E]  
  
[mrshcherbak@fedora ~]$  
[mrshcherbak@fedora ~]$
```

GPG keys

[New GPG key](#)

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.



Email address: `batyalove13@gmail.com`

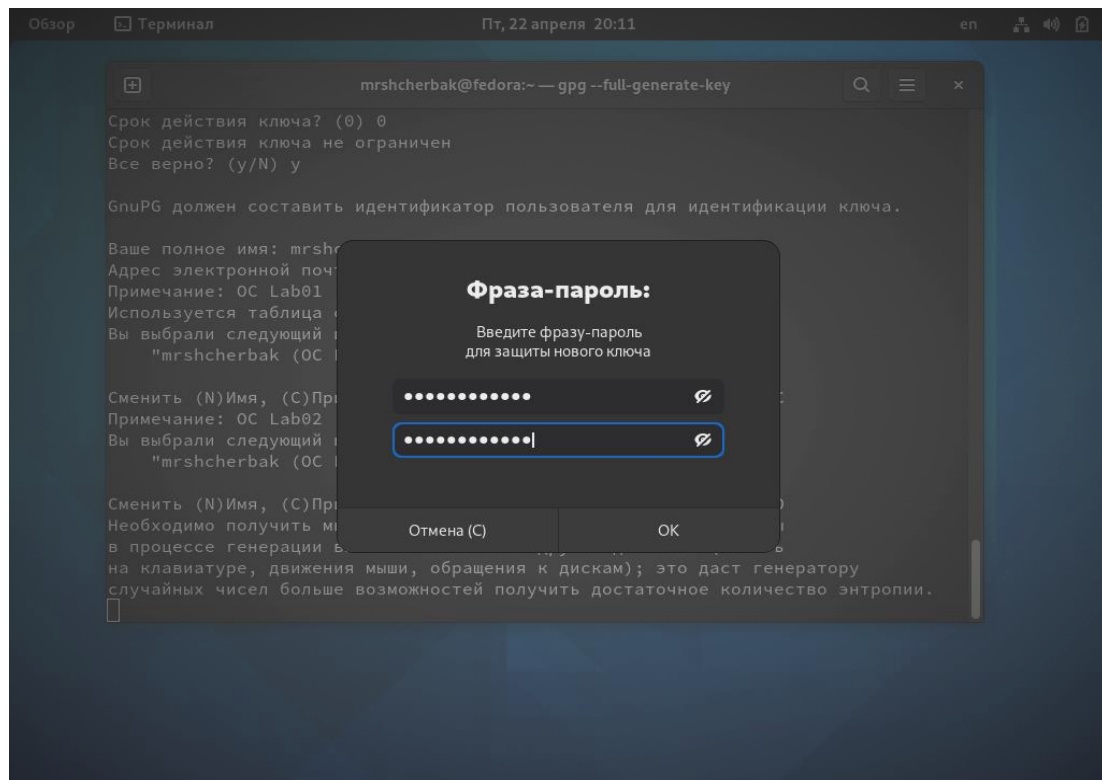
Key ID: `4D44943B3728F788`

Subkeys: `CF2F53C68C2F8633`

Added on 22 Apr 2022

[Delete](#)

Learn how to [generate a GPG key and add it to your account](#).



Настройка автоматических подписей коммитов git Рис.7

- Используя введённый email, указали Git применять его при подписи коммитов:
`git config --global user.signingkey`
`git config --global commit.gpgsign true`
`git config --global gpg.program $(which gpg2)`

Настройка gh

- Для начала необходимо авторизоваться
`gh auth login`
 - Утилита задаст несколько наводящих вопросов.
 - Авторизоваться можно через браузер.


```
https://github.com/login/device/success
mrshcherbak@fedora:~
[mrshcherbak@fedora ~]$ git config --global user.signingkey 4D44943B3728F788
[mrshcherbak@fedora ~]$ git config --global commit.gpgsign true
[mrshcherbak@fedora ~]$ git config --global gpg.program $(which gpg2)
[mrshcherbak@fedora ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
X Sorry, your reply was invalid: "Y" is not a valid answer, please try again.
? Generate a new SSH key to add to your GitHub account? Yes
? Enter a passphrase for your new SSH key (Optional) *****
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 3FE1-5B32
Press Enter to open github.com in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: /home/mrshcherbak/.ssh/id_ed25519.pub
✓ Logged in as MargaritaShcherbak
[mrshcherbak@fedora ~]$
```

Рис.7

Создание репозитория курса на основе шаблона Рис.8

– Создали шаблон рабочего пространства.

– Например, для 2021–2022 учебного года и предмета «Операционные системы» (код предмета os-intro) создание репозитория примет следующий вид:

```
mkdir -p ~/work/study/2021-2022/"Операционные системы"
```

```
cd ~/work/study/2021-2022/"Операционные системы"
```

```
gh repo create study_2021-2022_os-intro --template=yamadharma/course-directory-student-template --public
```

```
git clone --recursive git@github.com:/study_2021-2022_os-intro.git os-intro
```

```
mrshcherbak@fedora:~
--template=yamadharma/course-directory-student-template --public [mrshcherbak@fedora ~]$ git clone --recursive
git clone --recursive
git@github.com:<owner>/study_2021-2022_os-intro.git os-intro
fatal: Вы должны указать репозиторий для клонирования.

использование: git clone [<опции>] [--] <репозиторий> [<каталог>]

  -v, --verbose          быть многословнее
  -q, --quiet            тихий режим
  --progress             принудительно выводить прогресс
  --reject-shallow       don't clone shallow repository
  -n, --no-checkout      не переключать рабочую копию на HEAD
  --bare                 создать голый репозиторий
  --mirror               создать зеркало репозитория (включает в себя и параметр bare)
  -l, --local            для клонирования из локального репозитория
  --no-hardlinks         не использовать жесткие ссылки, всегда копировать файлы
  -s, --shared           настроить как общедоступный репозиторий
  --recurse-submodules[=<спецификатор-пути>]
                        инициализировать подмодули в клоне
  --recursive            alias of --recurse-submodules
  -j, --jobs <n>        количество подмодулей, которые будут клонированы параллельно
```

```
mrshcherbak@fedora:~/work/study/2021-2022/Операционные системы$ git clone --recursive git@github.com:MargaritaShcherbak/study_2021-2022_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Получение объектов: 100% (20/20), 12.50 КиБ | 2.08 МиБ/с, готово.
Определение изменений: 100% (2/2), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/mrshcherbak/work/study/2021-2022/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 42 (delta 9), reused 40 (delta 7), pack-reused 0
Получение объектов: 100% (42/42), 31.19 КиБ | 840.00 КиБ/с, готово.
Определение изменений: 100% (9/9), готово.
Клонирование в «/home/mrshcherbak/work/study/2021-2022/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 78, done.
```

Рис.8

Настройка каталога курса Рис.9

– Перешли в каталог курса:

```
cd ~/work/study/2021-2022/"Операционные системы"/os-intro
```

– Удалили лишние файлы:

```
rm package.json
```

– Создали необходимые каталоги:

```
make COURSE=os-intro
```

– Отправили файлы на сервер:

```
git add .
```

```
git commit -am 'feat(main): make course structure'
```

```
git push
```

```
mrshcherbak@fedora:~/work/study/2021-2022/Операционные системы/os-intro$ git push
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 78 (delta 31), reused 69 (delta 22), pack-reused 0
Получение объектов: 100% (78/78), 292.27 КиБ | 1.24 МиБ/с, готово.
Определение изменений: 100% (31/31), готово.
Submodule path 'template/presentation': checked out '3eaebb7586f8a9aded2b506cd1018e625b228b93'
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a1a7842a'
[mrshcherbak@fedora Операционные системы]$ cd ~/work/study/2021-2022/"Операционные системы"/os-intro
[mrshcherbak@fedora os-intro]$ rm package.json
[mrshcherbak@fedora os-intro]$ make COURSE=os-intro
[mrshcherbak@fedora os-intro]$ git add .
[mrshcherbak@fedora os-intro]$ git commit -am 'feat(main): make course structure'
[master 2829c1b] feat(main): make course structure
149 files changed, 16590 insertions(+), 14 deletions(-)
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
```

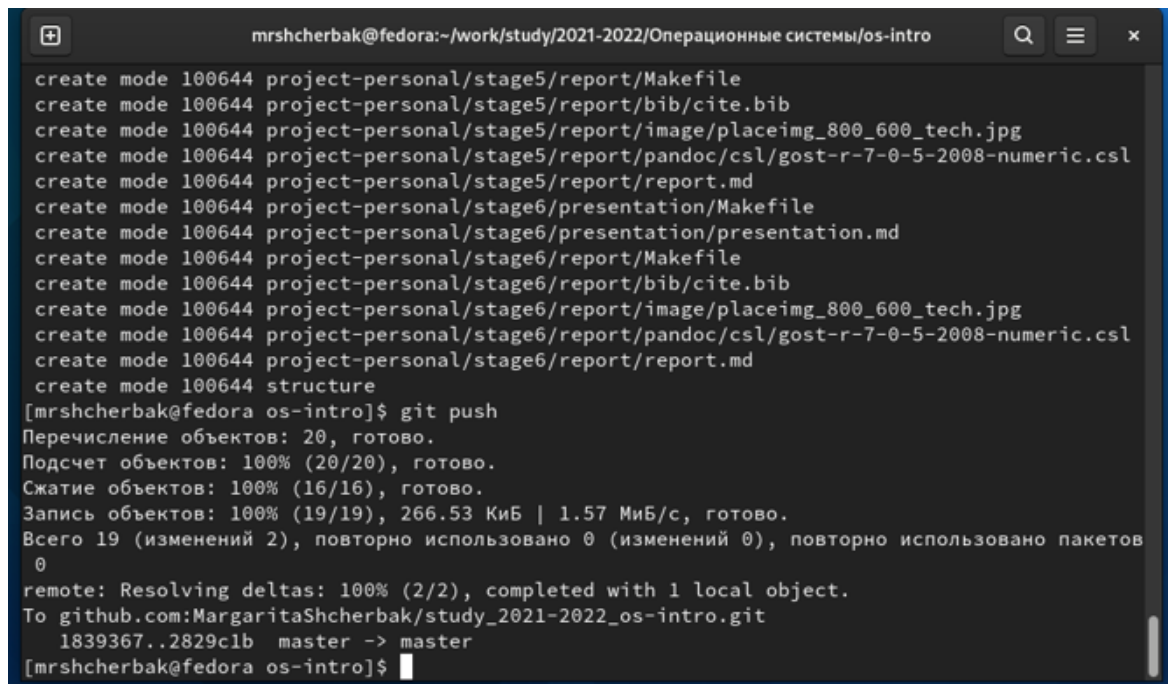
A screenshot of a terminal window with a dark background. The window title is 'mrshcherbak@fedora: ~/work/study/2021-2022/Операционные системы/os-intro'. The terminal shows a series of 'create mode' commands for various files in a project structure, including Makefiles, bib files, images, and CSL files. After these, the user enters 'git push'. The output shows the progress of pushing 20 objects, with 100% completion for each step. The final output indicates that the push was successful, with 1 local object being pushed to the remote repository 'MargaritaShcherbak/study_2021-2022_os-intro.git'.

Рис.9

Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Системы контроля версий -VCS- это программное обеспечение, которое используется для облегчения работы с изменяющейся информацией, обычно - в проектах. VCS позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое. Чаще всего используется при разработке, когда над одним проектом работает большое количество людей.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

- Хранилище (репозиторий) в системе контроля версий - это удаленный репозиторий, в котором хранятся все файлы проекта
- commit - фиксирует изменения перед загрузкой файлов в систему контроля версий
- история хранит все изменения в проекте, и при необходимости позволяет перейти в желаемое место

- рабочая копия - это копия проекта на компьютере разработчика. Если другой член команды изменил проект, вам необходимо скачать новую версию проекта на свой компьютер.

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд. В децентрализованных системах у каждого из участников проекта есть полная копия проекта на своем компьютере, что делает его менее зависимым от сервера (Git).

4. Опишите действия с VCS при единоличной работе с хранилищем.

Для начала необходимо создать и подключить удаленный репозиторий. Затем, поскольку никто, кроме вас, не изменяет проект, по мере изменения проекта отправляйте изменения на сервер, и нет необходимости загружать изменения.

5. Опишите порядок работы с общим хранилищем VCS.

Пользователь перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент.

6. Каковы основные задачи, решаемые инструментальным средством git?

Упрощение обмена информацией, ускорение разработки, устранение ошибок и недочетов при разработке.

7. Назовите и дайте краткую характеристику командам git.

- git init - инициализирует локальный репозиторий
- git add * или add. - добавляет файлы в репозиторий
- git commit - версия фиксации
- git pull - загружает текущую версию проекта
- git push - отправляет измененный проект на сервер
- git checkout - позволяет переключаться между ветками

- `git status` - текущий статус проекта
- `git branch` - просмотреть доступные ветки
- `git remote add` - добавить удаленный репозиторий

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.

`git push --all` (`push origin master/любой branch`)

9. Что такое и зачем могут быть нужны ветви (branches)?

Ветви функций, также иногда называемые ветвями тем, используются для разработки новых функций, которые должны появиться в текущих или будущих выпусках.

10. Как и зачем можно игнорировать некоторые файлы при `commit`?

Существуют временные и системные файлы, которые загромождают проект и не нужны. Путь к ним можно добавить в файл `.gitignore`, тогда они не будут добавлены в проект.

Вывод: таким образом, я изучила идеологию и применение средств контроля версий, освоила умения по работе с `git`, научилась использовать `Git`, и подключать удаленные репозитории, добавлять и удалять необходимые файлы, научилась использовать `Git Flow`, который значительно упрощает разработку проекта и навигацию между ветвями.